昵称：Nerxious
园龄：5年2个月
粉丝：534
关注：0
+加关注

< 2013年1月 >

日	一	二	三	四	五	六
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

搜索

找找看
谷歌搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

命令行(52)
mysql(21)
VIM(6)
io流(4)
XML(3)
Linux(2)
jdbc(1)
编码表(1)
反射(1)
泛型(1)
更多

随笔分类

Java(16)
Linux(3)
MySQL(21)
VIM(6)
命令行(52)

随笔档案

2013年5月 (3)
2013年1月 (61)
2012年12月 (33)

积分与排名

积分 - 151384
排名 - 1745

最新评论

1. Re:轻快的VIM（三）：删除
dd应该是剪切吧。当然，一般都拿来当成删除行。
--klous

2. Re:java中的匿名内部类总结
Thread t = new Thread() ...
t.start();

java中的匿名内部类总结

匿名内部类也就是没有名字的内部类

正因为没有名字，所以匿名内部类只能使用一次，它通常用来简化代码编写

但使用匿名内部类还有个前提条件：必须继承一个父类或实现一个接口

实例1:不使用匿名内部类来实现抽象方法

```
abstract class Person {
    public abstract void eat();
}

class Child extends Person {
    public void eat() {
        System.out.println("eat something");
    }
}

public class Demo {
    public static void main(String[] args) {
        Person p = new Child();
        p.eat();
    }
}
```

运行结果：eat something

可以看到，我们用Child继承了Person类，然后实现了Child的一个实例，将其向上转型为Person类的引用

但是，如果此处的Child类只使用一次，那么将其编写为独立的一个类岂不是很麻烦?

这个时候就引入了匿名内部类

实例2：匿名内部类的基本实现

```
abstract class Person {
    public abstract void eat();
}

public class Demo {
    public static void main(String[] args) {
        Person p = new Person() {
            public void eat() {
                System.out.println("eat something");
            }
        };
        p.eat();
    }
}
```

运行结果：eat something

可以看到，我们直接将抽象类Person中的方法在大括号中实现了

这样便可以省略一个类的书写

并且，匿名内部类还能用于接口上

实例3：在接口上使用匿名内部类

```
public class Demo {
    public static void main(String[] args) {
        Person p = new Person() {
            public void eat() {
                System.out.println("eat something");
            }
        };
        p.eat();
    }
}
```

运行结果：eat something

由上面的例子可以看出，只要一个类是抽象的或是一个接口，那么其子类中的方法都可以使用匿名内部类来实现

最常用的情况就是在多线程的实现上，因为要实现多线程必须继承Thread类或是继承Runnable接口

实例4：Thread类的匿名内部类实现

```
public class Demo {
    public static void main(String[] args) {
        Thread t = new Thread() {
            public void run() {
                for (int i = 1; i <= 5; i++) {
                    System.out.print(i + " ");
                }
            }
        };
        t.start();
    }
}
```

运行结果：1 2 3 4 5

实例5：Runnable接口的匿名内部类实现

```
1  public class Demo {
2      public static void main(String[] args) {
3          Runnable r = new Runnable() {
4              public void run() {
5                  for (int i = 1; i <= 5; i++) {
6                      System.out.print(i + " ");
7                  }
8              }
9          };
10         Thread t = new Thread(r);
11         t.start();
12     }
13 }
```

运行结果：1 2 3 4 5

分类: Java